International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Self-Organizing Maps-based Graph Convolutional Summarizer -based Multi-Model Approaches for Document Summarization

Abdulrahman Mohsen Ahmed Zeyad[1] and Arun Biradar[2]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: A brief overview of the same subject summarizes a lengthy text or paper. Most of the paper's crucial material must be retained while superfluous verbosity is eliminated. In instruction to produce a succinct summary for a document summarizing, the system collects keywords from papers or multiple documents. The basic idea is to limit or cut back on the quantity of crucial information in any given text. An information processing system that, given a collection of documents, extracts the essential information from the source while keeping the user or task in mind, then presents the summary in well-formed and concise prose and performs the assignment of a document summarizing. Summarizing numerous documents as opposed to only one is called multi-document summarization. The two main kinds are extractive and abstractive summaries of several materials. The most important and notable phrases and words from the original text are used to construct extractive resumes. However, some terms and sentences may not be found in the original text. This article focuses on ATS (Automatic document summarizing) methods that have recently been introduced. Deep learning-based models have recently been used for multi-document summarizing, which encourages the growth of text summarization and enhances model performance. We suggest the Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS). This extractive multi-document summary method uses SOM to ensure minimum performance constraints as an alternative to standard approaches. It fixes SMO-GCS's flaws and adds improvements that lead to a summarizer that enables phrase embedding and feature learning that is conscious of the graph structure. A rigorous methodology is needed to demonstrate how improvements are possible while still guaranteeing a minimal performance restriction. The effectiveness of the suggested summarizing approach is assessed using the DUC 2004 and Daily Mail/CNN datasets. The experimental findings show that SOM-GCS performs comparably to state-of-the-art summarization methods regarding ROUGE scores.

Keywords: *Deep learning, Document summarization, Automatic document summarizing, Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS), Multi-document summarizing*

1. Introduction

In today's rapidly advancing technology, analyzing and comprehending text files is challenging, time-consuming, and labor-intensive because of the exponential rise in data availability [1]. New, compelling text summarizing techniques must be developed to quickly and effectively process this volume of text data. An essential natural language processing (NLP) activity that can be helpful for many downstream applications, such as the construction of news digests, search engines, and report generation, is condensing a text or series of texts on the same topic, into a summary including crucial semantic information. Single-document summarizing (SDS) or multi-document summarizing (MDS) strategies can combine text from numerous sources into one document. While SDS is easier to use, it may not efficiently incorporate related or more recent studies, resulting in less thorough summaries. MDS, on the other hand, presents a challenge because it seeks to resolve potentially inconsistent and redundant information [2], producing more accurate and detailed summaries from documents produced at various times and from multiple perspectives.

Recently, deep learning has provided acceptable results in resolving several machine learning difficulties. It has demonstrated promising consequences in fields including speech recognition, sentiment analysis, and autonomous document summarization [3]. Methods of illustration learning with numerous layers of demonstration are known as deep learning. They are created by assembling basic, non-linear modules that elevate representations from one level to a problematic, marginally more abstract level. Deep understanding has various benefits, including the ability to apply to multiple computer science issues and the reduction of work needed for feature engineering [4]. Deep learning does not provide performance guarantees, despite its advantages. The cost function that is used to calculate costs during the learning process is shown in this study to be a convex function. As a result, a reduction can be finished in polynomial time, and empirical error can be decreased by selecting a suitable learning rate and performing enough rounds.

The need for enormous, labeled datasets is one of the main problems faced by many deep learning approaches, such as convolutional neural networks (CNNs). It takes a lot of time and money to create large datasets, and for various reasons,

[1]*Research Scholar, School of Computer Science and Engineering REVA University, Bangalore, India*
[2]*Professor, School of Computer Science and Engineering REVA University, Bangalore, India*
zeyadamam@gmail.com[1] and *arun.biradar@reva.edu.in*[2]

it frequently proves to be error-prone or even impossible. These label errors have been shown to occur even in often-used datasets for computer vision [5]. The logical (though a not necessarily simple) solution to these challenges is to construct deep learning models that can be trained on unlabeled/uncategorized data or to develop unsupervised learning methodologies for such deep networks. Numerous works that integrate or hybridize CNNs and self-organizing maps (SOMs) are congruent with this line of research. These models can either do the inverse and provide CNNs access to the unsupervised clustering capabilities of SOMs, or they can extract deep representations (like CNN codes) and quantize them into the SOM neural map [6].

Machine learning algorithms are taught to comprehend all available data and filter the relevant information. Machines can easily compress large amounts of data; doing so manually would be difficult, expensive, and time-consuming. But the next issue is choosing the most excellent significant data from the main manuscript and then condensing the data that has been picked out. The end product, or summary, is given more attention in research on document summarization than the reasons why the text should be understood [7]. A more profound comprehension of the cognitive underpinnings of the activity would be beneficial for addressing some inadequacies in present systems. While earlier studies on summarization concentrated on the summation of a single text, current methodologies frequently focus on the summary of numerous documents.

As seen in Fig. 1, any language-specific text document can be used as input, including news items, legal documents, medical documents, and other report materials.

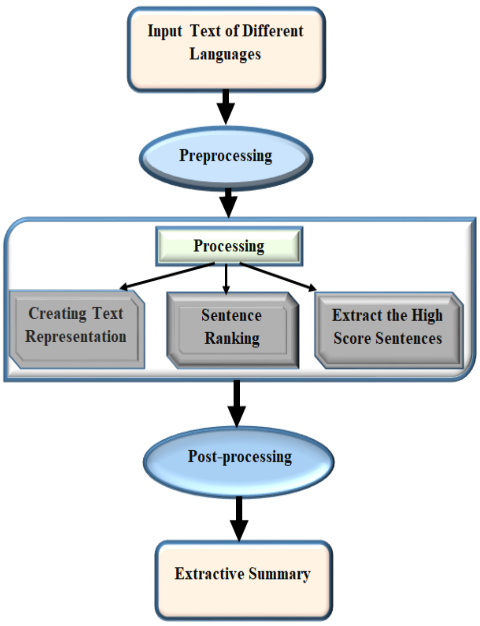

Fig 1: General structure of Automatic Document Summarization

Any ATS system created in a different language should have as its primary goal the automatic creation of a summary from the input text document or documents that are shorter than the original document(s), contains only the most crucial information, and leaves out any irrelevant information. Automatic document summarization (ATS) can be performed using a variety of methods, but for the most part, we focus on two: extraction-based and abstraction-based methods. Even though the extraction-based summary is more straightforward, abstraction-based summarization is preferred. Without altering the original text, the extraction-based technique summarizes all the essential information from the primary source [8]—However, the abstraction-based approach functions similarly to how people behave. The original text is modified, adding new phrases, resulting in a grammatically accurate summary.

As a result, the abstraction strategy summarizes information more effectively than the extraction approach. The extraction approach is more widely used since text outlining algorithms cannot be easily created [9]. In the past, extractive summarizers frequently used sentence scoring to extract the summary. However, there are already several cutting-edge techniques for producing summaries that use the linguistic or statistical assets of the manuscript, such as common keywords, high-frequency terms, cue methods, position methods, and title approaches for determining sentence weight ages.

This article focuses on Automated Document Summarization (ATS) methods that have only recently been used. Relatively lately, deep learning-based representations have been utilized for multi-document summarizing, fostering the expansion of document summarization and progressing the overall presentation of pictures. The Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS), a technique for the extractive multi-document summary that uses SOM to ensure a slight performance limitation, is an alternative to the standard approaches. It corrects the shortcomings of SOM-GCS and makes improvements that result in a summarizer that enables phrase implanting and feature learning that is aware of the graph's structure. To demonstrate how improvements can be made while still guaranteeing a minimal performance restriction, an approach that is both rigorous and comprehensive is required. The DUC 2004 and Daily Mail/CNN datasets are used in this evaluation to help determine whether the suggested summarizing method is effective. The results of the experiments indicate that SOM-GCS achieves ROUGE score results comparable to those achieved by state-of-the-art summarization approaches.

1.1 Contributions

- In new times, deep learning-based models for multi-document summarizing have been utilized, promoting document summarization development and improving model performance.

- This paper suggests the Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS). This extractive multi-document summary technique uses SOM to ensure minimum performance constraints as an alternative to standard approaches.
- It fixes SOM-GCS's flaws and adds improvements that lead to a summarizer that enables phrase embedding and feature learning that is conscious of the graph structure.
- Finally, to demonstrate the viability of our suggested approach, we carried out comprehensive experiments. We present our findings using the DUC 2001 and 2002 multi-document summarization datasets as our benchmark. The proposed method outperforms all current methods compared to all measure metrics, and the considerable results support the method's applicability for opinion-oriented summarization.

The respite of the essay is organized as surveys. The next unit provides a summary of recent relevant studies from Section 2. Section 3 covers our suggested technique, RLDS, in more depth. Section 4 of our experimental findings explain them. Lastly, in Section 5, we offer our consequences and recommend additional investigation.

2. Literature Survey

According to Reinald et al. [10], extractive-abstractive models cannot easily construct specific summaries based on preferences, and their summaries are less accurate and useful. To overcome these limitations, they projected a two-stage Condense-Abstract (CA) paradigm by employing opinion summarization as an example of multi-source transduction. The condense model's Bi-LSTM auto-encoder uses each input document to learn representations at the word and document levels. The Bi-LSTM's hidden states in both directions are concatenated as a word-level representation. The initial and last word encodings make up the document representation. The abstract model uses a straightforward LSTM decoder, a standard attention mechanism, and a copy mechanism to create a summary of the user's ideas.

Abhishek et al. [11] created an extractive MDS paradigm considering document-dependent and document-independent data. Using a CNN with several filters, this technique obtains phrase-level representation. The proposed Bi-LSTM tree indexer is fed full binary trees created with these salient illustrations to improve generalization skills. Leaf node modification employs an MLP with a ReLU function. To internment both semantic and compositional information, the Bi-LSTM tree indexer, in particular, combines the time serial power of LSTMs with the compositionality of recursive models.

Logan et al. [12] created a two-stage summation approach that considers semantic compatibility because most instant verdicts are created by merging one or two source words. This technique uses combined scores on individual sentences and sentence pairs to filter representative sentences from the source materials. High-scoring sentences or sentence pairs are condensed and modified to create a summary that utilizes the PG network. This research uses an indiscriminate Transformer-based model to encode single phrases and sentence pairs to obtain a rich contextual illustration of words and sequences.

Bahloul et al. [13] introduced a summarizer unsupervised hybrid technique-based automatic system for Arabic single-document summaries. This combines a cluster-based, statistical-based, and graph-based technique. They separated the text into subtopics, chose the most pertinent sentences from those subtopics, and then applied a selection algorithm to a graph representing various lexical and semantic links between phrases. The Essex Arabic summaries corpus was used for simulation. It was compared to merged model graphs, autonomous summarization engineering, and n-gram graph powered by regression, demonstrating that the recommended method outperforms the others.

According to Mokhale et al. [14], summarizing the content of numerous papers is essential to extract relevant information. Due to the massive rise in range requiring only a summary to be retrieved quickly, the authors also reviewed several strategies for summarizing numerous documents. Document metadata were combined by Saeed et al. [15], who also used multistage clustering to evaluate unstructured documents. To connect the created clusters, adjacency graphs are produced. Authors undertake multistage clustering and interlinking using sub-corpuses. The authors processed six alternative metadata combinations over text queries using their methodology on a new data set, yielding 67% associated text. The SHAP (Shapley Additive exPlanations) model assesses this method.

In the paradigm presented by Dima Suleiman et al. [16], one layer was employed at the decoder, and two layers (input text layer and name entities layer) were used at the encoder. The encoder and the decoder employ LSTM, although the former employs a bidirectional LSTM, while the latter utilizes a unidirectional LSTM. We used a word embedding model trained with the AraVec package. We used a dataset collected and cleansed in advance to be suitable for abstract summarization for this test. The scores for ROUGE1 and ROUGE1-NOORDER, employed in the evaluations, were 38.4 and 46.4, respectively. The produced dataset, however, is minimal and not publically accessible. Thus it cannot be utilized for cross-study comparisons.

Ukan et al. [17] developed a Maximum Independent Set-based graph-based multi-document summarizing method. This procedure consists of three steps. Stop-words are cut out in the beginning. In the second stage, which entails mathematical modeling of the word similarity between the phrases, the sentences consistent with the nodes in the Maximum Independent Set are eliminated from the primary network. After the terms that make up the articles are given a weighted average using the eigenvector node centrality technique, the most excellent significant sentences are picked to make up the instantaneous.

Jin et al.'s [18] additional recommendations included a Transformer-based multi-granularity interaction network and a combined extractive and abstractive MDS. The three granular layers are words, sentences, and documents, constituting a semantic unit. A network of granular hierarchical relations links these levels together. The semantic linkages are captured at the same level of detail using a self-attention approach. The extractive summary uses sentence granularity representation, while the abstractive summary uses word granularity. A sparing attention method is also employed to ensure that the summary generator concentrates on crucial information.

Li et al. [19] industrialized an RNN-based system to extract significant information vectors from phrases in contribution documents automatically. Cascading attention can rebuild the innovative contribution sentence vectors by keeping the most relevant embeddings. The suggested approach uses a sparsity requirement to punish unnecessary information in the output vectors during reconstruction.

Cao et al. [20] created a TCSum perfect with an extra text classification sub-task integrated into MDS to provide more supervision signals. The text classification model uses the CNN descriptor to develop documents onto the distributed demonstration and categorize contribution documents into various groups. The TCSum ideally selects the relevant category-based alteration conditions in line with the classification results to translate the predictable sentence embedding from the organization model into the summary embedding.

Zhuang et al. [21] proposed a statistical and probabilistic technique for query-based summarization that was presented to uncover various topics from a considerable document collection. The query-specific knowledge is integrated into the topic-based module by constructing the generative topic modeling technique. As a result, a module that accurately approximates the query is created, and themes related to particular documents are created. Second, topic-based models are built by regularizing query-specific data. Though the generated summaries are highly relevant, the system frequently needs to identify the relationship between the query and the source document.

Zhao et al. [22] address a significant issue concerning summary redundancy. The self-adaptive differential Evolution technique makes the output summary more convergent and less redundant. The algorithm is similar to a genetic algorithm in that it randomly chooses a group of individual phrases from the decision space. When it is necessary to maximize the generic summary with the least amount of duplicated information, self-adaptive differential evolution is applied. However, the drawback of increased run time complexity is more significant.

Ayetiran et al. [23] This article explains how to rate texts using Recursive Neural Networks (RNN), presented throughout the piece. The hierarchical regression approach is used to identify the order in which the statements in this part should be submitted to the reader. The input is provided in the form of rules crafted by hand and delivered to the system. After that, the RNN will automatically learn the pattern of words that leads to the building of sentences with the assistance of the supervision provided by the parse tree.

Mallick et al. [24] proposed that the objective functions are more complex here, making them tougher to handle, and that the user's opinion is also considered. In such circumstances, conventional reasoning algorithms did not process medical machine-learning records. However, they were commonly utilized in many machine learning applications because the information from short and partial data samples could not be found. An external rule base is used to extract relevant data from biological data. They applied swarm intelligence, a versatile and promising machine learning technique. Another helpful technology for machine learning is Particle Swarm Optimization (PSO), which is based on a qualitative rule base and gives consistent previous findings.

Alguliyev et al. [25] presented COSUM, a two-step sentence assortment approach based on clustering and optimization, to construct extractive summaries of input documents. In this model, sentence collections from a record were divided depending on the subjects using the K-Means technique. The number of terms shared by sentences was utilized to determine how similar the phrases were. Specific sentences from each cluster were chosen to utilize optimization to maximize coverage and diversity in the chosen verdicts. The ROUGE-1 and ROUGE-2 measures, coupled with the paper summarizing datasets from DUC 2001 and DUC 2002, were used to assess COSUM. LexRank, Conditional Random Field, Manifold ranking, and DE-based techniques were used to evaluate COSUM's performance.

2.1. Limitations of Existing system

- Limited availability of multimodal data: Multi-model techniques for document summarization necessitate access to vast amounts of multimodal data, such as text, audio, photos, and videos. However, such information is only sometimes available for all sorts of documents, and obtaining it can be complicated and costly.
- High computational costs: Multi-model techniques often necessitate more excellent computational resources than traditional text-based approaches. The various modalities must be processed separately before being integrated to produce the final summary. This can be computationally time-consuming.
- Difficulty in choosing appropriate modalities: Choosing the most relevant modalities for a text might be difficult. This is because different modalities may include other information, and it can be difficult to discern which modalities are most relevant for summarizing a particular document.
- Lack of interpretability: Multi-model approaches can be more complicated than text-based approaches. This is because the many modalities may contain different types of information, and it cannot be easy to understand how the various modalities are merged to get the final summary.
- Difficulty in evaluating performance: Evaluating the performance of multi-model techniques can take time and effort. This is because there is yet to be a commonly established standard for measuring the success of multi-model approaches to document summarization, and it can be challenging to determine which metrics are most appropriate.

2.2. Problem Identification of Existing system

- Document summarization, a crucial task in natural language processing, condenses long documents into concise summaries while preserving essential information. To construct an outline for a given paper, multi-model techniques for document summarization employ various models, such as extractive and abstractive summarization. Despite their potential benefits, multi-model systems confront several problems that must be addressed.
- One of the most significant difficulties is identifying the optimum model combination for a given document. Different documents may require other models; determining the appropriate mix of models to provide a high-quality summary can be tricky. Furthermore, integrating numerous models can be complicated, and combining their results into a coherent summary might require much work.
- Another area for improvement is the need for large-scale datasets for testing multi-model techniques. Existing datasets are frequently limited in size and may not reflect the diversity of documents and languages. As a result, determining the efficacy of multi-model techniques across domains and wording can be difficult.
- Furthermore, multi-model approaches may require more computational resources than single-model approaches, limiting their practical application. Finally, multi-model techniques may be challenging to read, which limits their utility in situations where interpretability is crucial, such as the legal or medical domains.

3. Proposed System

This section concerns recently developed ATS (Automatic document summarizing) approaches. Recent advances in text summarization have been made possible by using deep learning-based models for multi-document summarizing. This improves model performance and stimulates the growth of text summarization. We propose the Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS) as an alternative to traditional methods. This extractive multi-document summary technique uses SOM to guarantee minimum performance restriction. The shortcomings of SMO-GCS are addressed, and new enhancements are introduced, resulting in a summarizer that supports phrase embedding and feature learning while considering the graph structure. A rigorous methodology is required to show how enhancements are feasible while guaranteeing a low-performance restriction.

3.1. Multi-document text summarization

The popularity of making text content available online, such as public news, has brought attention to the utility of text summarizing software. Several advantages of automatic text summarizing include decreased study time and improved indexing performance. Extractive and abstractive techniques are the two primary ways to produce involuntary reviews. The former generates a summary using natural language generation techniques, while the latter chooses a subsection of verdicts from the original text(s). One text or a collection of papers can be the input to a summarizer. This study's primary focus is extractive multi-document summarization. It is possible to describe automatic extractive multi-document summarization officially as surveys. $dt_1, dt_2, ..., dt_n$ are text documents on a particular topic and $DT = \{dt_1, dt_2, ..., dt_n\}$ a set of papers. Respectively text di (for i [1, n]) is made up of a collection of sentences called $dt_i = \{st_{1,i} st_{2,i}, ...st_{m_i,i}\}$, where

$m_i = |d_i|$ denotes how many sentences are included in each document di. The objective of the multi-document summarizing is to create an instant $ST = \{st_1, st_2, ..., st_{\square}\}$ by choosing informative and non-redundant phrases from $V = \cup_{i=1}^{n} \{st \mid st \in dt_i\}$ under constraint C (similar to the cardinality constraint). Let's assume that all the words in D are part of a more extensive set, or context, denoted by V. The preceding reasoning suggests that determining the best instant can be viewed as a combinatorial optimization problem, which is proved to be NP-hard.

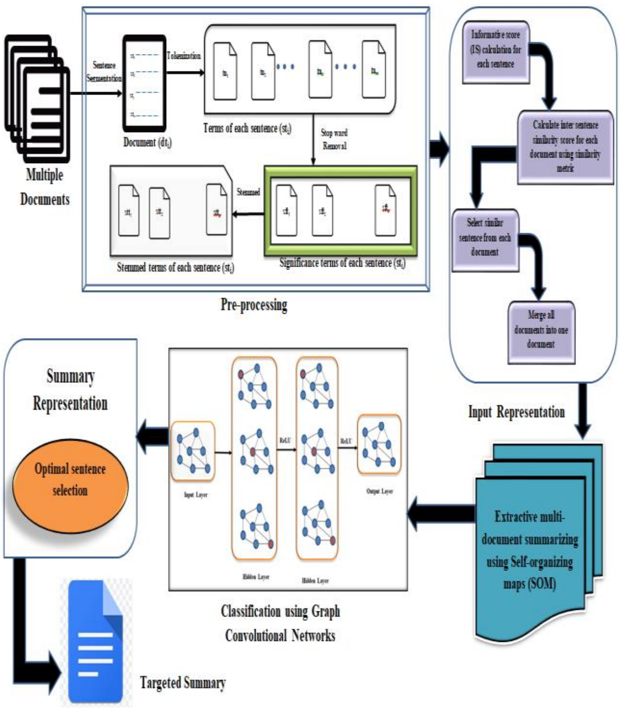

Figure 2: Proposed method of SOM-CGS

Using sub modularity in the setting of artificial manuscript summarization is promising, as shown by the tests provided in the next section. Additionally, as was already established, sub modularity is inherent in this situation. As a result, we make an effort to explain extractive multi-document summarization using SOM-GCS and concepts that are linked to it: Summary S* will be produced by maximizing f (K) SOM-GCS under the cardinality constraint C. Extractive multi-document summarization, or Eq. (1), is the formulation for this process.

$$S^* = \arg\max_{S \subseteq C} \{ f_{DSN}^{(K)}(ST) \} \ \ s.t. \ \ \sum_{st \in ST} C(st) \leq \square \tag{1}$$

Where C(.) denotes the time and effort needed to change a particular sentence from its past tense to the present, one way to regulate a sentence's price is to count its arguments. Fig. 2 displays the block diagram for the SOM-GCS technique.

3.1.1 Pre-processing

There are four sub processes in pre-processing.

- **Sentence segmentation:** Each document D from the collection of input text documents is segmented independently as $dt = \{st_1, st_2, ...st_n\}$ it st_j stands for the document's j^{th} sentence to make it easy to extract the immediate sentence, and n is the total amount of sentences.
- **Tokenization:** The standings of each sentence are denoted by $TN = \{tn_1, tn_2, ...tn_m\} =$, where T is the total number of characters and $k = 1, 2, ..., m$ is the set of all the unique situations in D.
- **Stop words are dropped:** Words like "a," "an," and "the," which are frequently used in English but have no bearing on the paper, are eliminated.
- **Stemming:** To generate a more general base form, this technique includes clipping off the ends of words [26].
- **Named entity recognition (NER):** This locates every textual mention of the named entities in texts and classifies them into predefined interest categories (such as a person, place, or organization). In a text, NER specifically labels word sequences that are names of items. In our model, NER determines how relevant sentences are as having important "entities" to produce the final ranking.

3.1.2 Input Representation

A weight (sum of period occurrences), the informative sentence score, is calculated for each sentence using the input word form data. The implementation also involves providing the optimization algorithm with a sentence weight corresponding to the sentence's informative score.

3.1.3 Extractive multi-document summarizing using SOM

3.1.3.1 The SOM Algorithm

The SOM produces a discrete topological mapping of input space $Y \in \square^n$ using a group of neurons frequently arranged in a 2-D rectangular or hexagonal grid. All weights were initialized to minimal arbitrary statistics at the beginning of the learning process. Let $\{v_1, v_2, ..., v_M\}$ be the vector displaying the neuron i's grid position. The weight vector Wi is connected to neuron i and shares the exact dimensions as the input vector n and the entire amount of neurons M. The processes listed in Algorithm 1 are then repeated by the algorithm, where Ω is the set of neuron indexes and

$\eta(w,k,u)$ is the neighborhood function. A Gaussian form of the neighborhood function—more specifically,

$$\eta(w,k,u) = \exp\left[-\frac{\| r_w - r_k \|^2}{2\sigma(u)^2}\right] \qquad (2)$$

It is frequently used in practice, even though one can still use the unique stepped or top-hat kind (one when the neuron is inside the neighborhood; zero otherwise). It stands for the neighborhood's effective range, frequently shrinking over time.

Algorithm 1 Self-Organizing Map algorithm

repeat

1. At each time u, present an input $y(u)$*, and select the winner,*

$$w(u) = \arg\min_{k\in\Omega} \| y(u) - v_k(u) \| \quad (9)$$

2. Update the weights of the winner and its neighbours,

$$\Delta v_k(u) = \alpha(u)\eta(w,k,u)[y(u) - v_w(t)] \quad (10)$$

until the map converges

The 'learning rate' or 'adaptation gain' coefficients denoted by $\{\alpha(u), u \geq 0\}$ are scalar-valued, monotonically decreasing, and fulfill [27]:

$$(i)\ 0 < \alpha(u) < 1$$

$$(ii)\ \lim_{t\to\infty} \alpha(u) \to \infty$$

$$(iii)\ \lim_{u\to\infty} < \alpha^2(t) < \infty \qquad (3)$$

The ones employed in stochastic approximation are the same ones that they are. Ritter and Schulten substituted the less stringent $\lim_{u\to\infty} \alpha(u) \to 0$ for the third criterion in Equation (3).

If the best matching rule is implemented using the internal product similarity measure,

$$w(u) = \arg\min_{k\in\Omega}[w_k^U y(u)] \qquad (4)$$

Therefore, the updated weight will correspond to

$$v_k(u+1) = \begin{cases} \dfrac{v_k(u) + \alpha(u)y(u)}{\| v_k(u) + \alpha(u)y(u)} & k \in \eta_w \\ v_k(u) & k \notin \eta_w \end{cases} \qquad (5)$$

Text/document mining applications frequently employ this type of form.

The SOM algorithm maintains topology while clustering or vector-quantizing the input space and building a map. In the past, it has also been employed for classification. In this case, data from well-known categories train the map. The nodes are then labeled or categorized to enable the map to classify samples that cannot be seen.

3.1.4 Utilizing Graph Convolutional Summarizer Networks for Classification

On top of the phrase relation graph, we employ [28] Graph Convolutional Networks (GCN). This section details the history of GCN and how the network generates the last sentence embeddings. Now, let's talk quickly about the layers of how the GCN spreads, as shown in Fig. 3.

GCN's objective is to train a purpose $f(Y,B)$ that accepts the following inputs:

- $B \in \mathbb{R}^{N\times N}$, the graph's adjacency matrix, where N is the entire quantity of nodes in the network.
- D, the dimension of the contribution node feature vectors, and $Y \in \mathbb{R}^{N\times D}$, the input node feature matrix.

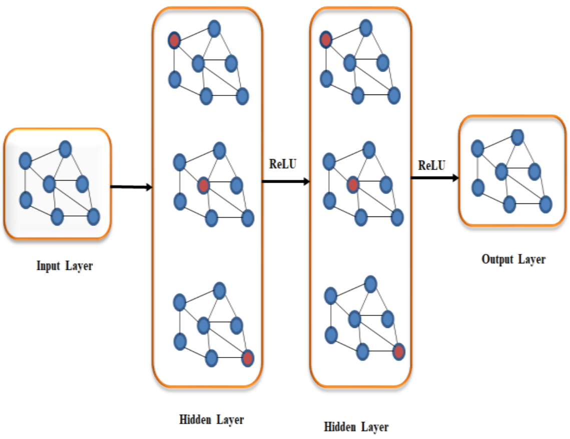

Figure 3: Structure of Graph Convolutional Networks

For each node, $Z \in \mathbb{R}^{A\times F}$ it makes high-level hidden topographies that capture the graph's structure. The letter F stands for the output feature vectors' dimension. The function $f(Y,B)$ employs neural network-based layer-wise propagation. Starting with H0 = X, we construct the activation matrix in the $(l+1)^{th}$ layer as $H^{(l+1)}$. $Z = f(Y,B) = H^{(L)}$ is the output of L-layer GCN. Consider a simple type of layer-wise propagation to introduce the formulation:

$$H^{(l+1)} = \sigma(BH^{(l)}V^{(l)}) \qquad (6)$$

Where is a function like $\mathrm{Re}LU(.) = \max(0,.)$ that activated. The l^{th} layer's learning parameter is called $V(l)$. Eq 6 has two issues. First, we add the feature vectors of all nearby nodes, not the node itself, as shown by dividing by B. This is fixed by adding self-loops to the graph. Secondly, multiplying a feature vector by B changes its scale since B is not normalized. To address this, we use symmetric normalization $D^{-\frac{1}{2}}BD^{-\frac{1}{2}}$, where D is the node degree matrix. The following propagation rule results from these two renormalization tricks:

$$H^{(l+1)} = \sigma(D^{-\frac{1}{2}}AB^{-\frac{1}{2}}H^{(l)}V^{(l)}) \tag{7}$$

Where $B = B + I_N$ is the adjacency matrix of graph G with additional self-loops (IN is the identity matrix) The degree matrix $D_{ii} = \sum_j B_{ij}$ is called D. Furthermore, Eq 7 is supported by [28] as a first-order estimate of spectral graph convolution.

For instance, if we have a two-layer GCN, we compute $B = D^{-\frac{1}{2}}BD^{-\frac{1}{2}}$ during the pre-processing stage before generating [29-30].

$$Z = f(Y,B) = \sigma(B\sigma(BYV^{(0)})V^{(1)}) \tag{8}$$

3.1.5 Summary Representation

The goal of summary demonstration is to create summaries of document sets with valuable data. The best sentence collection approach compares the informative sentence score produced by the optimization algorithm with deference to a predetermined inception assessment to choose the key sentences that serve as the summary.

3.2. Summary Evaluation Criteria

The document summarization problem aims to provide as informative a summary as possible while minimizing redundancy and maintaining readability. To create the most significant outline possible, the authors of this study attempted to summarize document sets utilizing a variety of aims, including satisfied coverage as well as non-redundancy, cohesion, and readability. These objectives are clarified in the impartial meaning $m(S)$ and formalized as three sub-functions,

$m_{\mathrm{cov}}(S), m_{\mathrm{co}h}(S), m_{read}(S)$.

$$m(S) = m_{\mathrm{cov}}(S) + m_{\mathrm{co}h}(S) + m_{read}(S) \tag{9}$$

Each sentence's content coverage, in summary, is shown as follows:

$$m_{\mathrm{cov}}(S) = Sim(s_i, P) \quad i = 1,2,..,n \tag{10}$$

Where O = the center of the foremost satisfied group of sentences, i.e., $P = \{P_1, P_2, ...P_n\}$ of document sets, and P_i the weighted average of the sentences in each document. The comparison among and O is assessed to establish the relevance of the sentences. High content coverage is reflected in higher similarity values. The following diagram illustrates how the sentences, in summary, are related:

$$m_{\mathrm{co}h}(S) = 1 - Sim(s_i, s_j) \ where \ i \neq j \ and \ i, j = 1,2,..n \tag{11}$$

Utilizing several diverse document sets is necessary for joining concepts at the sentence and subsection levels. This improves the reader's ability to understand the entire content. Therefore, a more significant $m_{\mathrm{co}h}(S)$ value indicates a stronger relationship between phrases and vice versa. The similarity between the two sentences as determined by summary readability is:

$$m_{read}(S) = Sim(s_i, s_j) \ where \ i \neq j \ and \ i, j = 1,2,..n \tag{12}$$

The comparison s_i among s_j is measured by $m_{read}(S)$. A more excellent value indicates the readability of the summary $m_{read}(S)$.

4. Result and Discussion

4.1. Experiments

Experiments were run on a few benchmark datasets to gauge the SOM-GCS's performance. This section describes the datasets, evaluation measures, and implementation specifics. In addition, the study results and summary parameters are provided. It's critical to remember that stop-words are deleted in all experiments. Additionally, the sentence situation feature is not used in developing SOM-GCS because of its iffy association with label scores in the training set.

4.2. Datasets

We used the DUC 2004 and Daily Mail/CNN datasets in our experiments. The SOM-GCS model was developed with the aid of DUC 2004. In DUC 2004, there were a variety of tasks for various jobs, including question-and-answer sessions and general multi-document summaries. The position summaries in these two datasets have been converted into their extractive equivalents using the technique discussed in the preceding units because SOM-

GCS should be trained in an extractive fashion. Still, they have generated abstractly. Using the DailyMail/CNN dataset, we evaluated SOM-GCS's performance compared to cutting-edge single-document summarizers. Using the DailyMail/CNN dataset, we assessed SOM-GCS's performance compared to cutting-edge single-document summarizers.

4.3. Implementation

The summarizer was implemented using the TensorFlow package. Using DUC scripts2 and the CoreNLP3 package, text pre-processing tasks such as sentence segmentation, stemming, tokenization, and stop-word removal were completed. Per cluster, documents context graphs with nodes representing pre-processed phrases and edge weights that showed how related they were based on the Jaccard index were built. A method was used to provide more training items to SOM-GCS to increase the dataset samples. The dataset was mapped into numerical normalized vectors using significance and effect attributes in this process. The SOM-GCS was then trained using the retrieved vectors as input.4

4.3.1. Precision Analysis

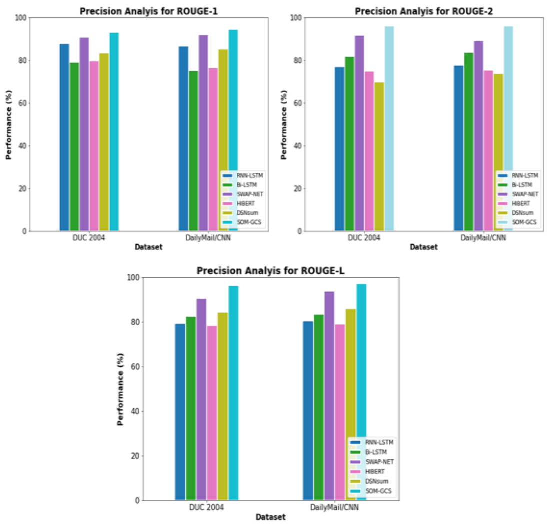

Fig 4: Precision Analysis of SOM-GCS for ROUGE-1, ROUGE-2 and ROUGE-L

A precision comparison of the SOM-GCS methodology with other known methods is shown in Fig.4 and Tab.1. The graph shows that SOM-GCS for ROUGE-1 has an increased efficiency with precision. For data set DUC 2004, for instance, SOM-GCS has a precision of 93.06%, whereas SWAP-NET method has a precision of 90.75%, and Bi-LSTM has a precision value of 78.97% respectively with first, second and last positions. However, the SOM-GCS model performed best with varying data sizes. Similarly, with DailyMail/CNN dataset, the SOM-GCS has a precision of 94.56%, SWAP-NET model has a precision of 91.87% and Bi-LSTM has a precision of 75.05%. Similarly, with ROUGE 2, the proposed method SOM-GCS has a Precision of 96.05% for DUC 2004 and 94.56% for DailyMail/CNN dataset. For ROUGE-L, the proposed SOM-GCS resulted in improved performance with precision for different dataset. While the precision values for the RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models are, respectively, 79.34%, 82.54%, 90.45%, 78.34%, and 84.32%, the SOM-GCS model has demonstrated maximum performance with 96.25% for the DUC 2004 data set. With DailyMail/CNN data set the SOM-GCS model has shown maximum performance with the precision of 97.32%, while it is 80.54%, 83.43%, 93.76%, 79.03%, and 85.96% for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models respectively.

4.3.2. Recall Analysis

The SOM-GCS methodology is contrasted with other accessible methods in Fig. 5 and Tab. 2 in terms of recall. The graph shows that ROUGEs' performance with recall were enhanced with SOM-GCS. For instance, SOM-GCS has a recall value of 91.28% for the data set DUC 2004, while Bi-LSTM models have a recall value of 89.34%, the second-highest number, while DSNsum has a recall value of 74.21% the lowest performance. However, the SOM-GCS model performed best with varying data sizes. Similarly, with DailyMail/CNN dataset, SOM-GCS has a recall value of 92.89%, whereas Bi-LSTM models have a recall of 89.45%, the second highest number, and RNN-LSTM has a recall of 80.43%, which is the lowest value for ROUGE 1. Similarly, with ROUGE 2, the proposed method SOM-GCS has a Recall of 93.87% and 94.76% for DUC 2004 and DailyMail/CNN dataset respectively, while it is between 75.90% and 89.78% for the other existing methods for both DUC 2004 and DailyMail/CNN dataset. For ROUGE-L, the recall value of the SOM-GCS model is 95.89%, compared to 76.93%, 89.12%, 82.34%, 86.43%, and 77.93% for the RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models, respectively with DUC dataset. And with DailyMail/CNN data set, the SOM-GCS model has shown maximum performance of recall values with 96.19%, while it is 82.34%, 90.65%, 85.01%, 89.64%, and 80.14% for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models respectively.

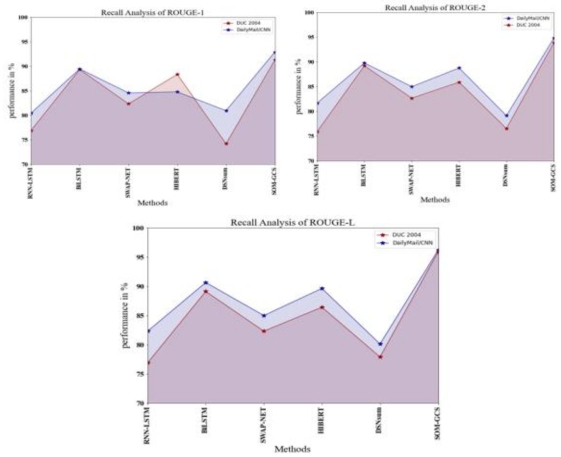

Fig 5. Recall Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

4.3.3. Classification Accuracy

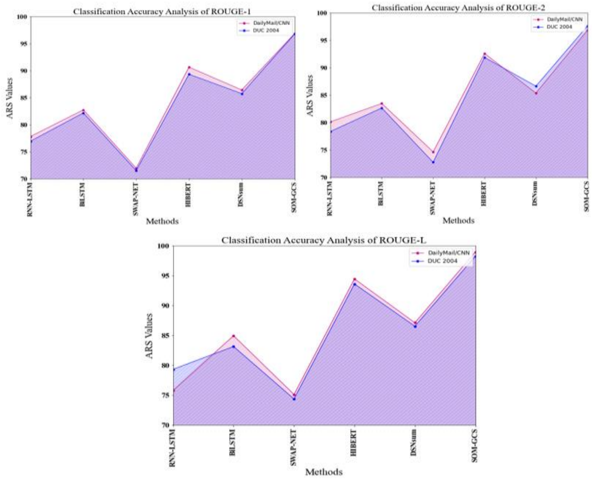

Fig 6: Classification Accuracy Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

A comparison of the SOM-GCS methodology's classification accuracy against other known methods is shown in Fig. 6 and Tab. 3. The graph shows how SOM-GCS has an enhanced performance with classification accuracy for ROUGEs. Using the data set DUC 2004, for instance, SOM-GCS has a classification accuracy of 96.76%, while HIBERT models come in second with a value of 89.34%, and SWAP-NET comes in last with a value of 71.54%. However, the SOM-GCS model performed best with varying data sizes. Similarly, with DailyMail/CNN dataset, SOM-GCS has a classification accuracy of 96.85%. In contrast, HIBERT models have a classification accuracy of 90.65%, the second highest number, and SWAP-NET has a classification accuracy of 71.98%, the lowest value for ROUGE 1. Similarly, with ROUGE 2, the proposed method SOM-GCS has a classification accuracy of 97.56% and 96.79% for DUC 2004 and DailyMail/CNN dataset respectively, while it is between 72.78% and 92.59% for the other existing methods for both DUC 2004 and DailyMail/CNN dataset. For ROUGE-L the classification accuracy of the SOM-GCS model is 98.24%, whereas it is 79.34%, 83.15%, 74.38%, 93.61%, and 86.54% for the RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models, respectively with the DUC 2004 data set. And with DailyMail/CNN dataset SOM-GCS model has shown maximum performance for a classification accuracy of 98.95%. In comparison, it is 75.83%, 84.95%, 75.13%, 94.45%, and 87.12% for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models correspondingly.

4.3.4. F-Score Analysis

The SOM-GCS methodology is compared to other available methods for Fscore analysis in Fig. 7 and Tab. 4. The graph shows how SOM-GCS for ROUGEs has an enhanced performance with fscore. SOM-GCS, for instance, has a Fscore of 94.12% for the data set DUC 2004, while HIBERT models, have a Fscore of 90.56%, which is the second-highest value, while SWAP-NET has a fscore of 76.13%, which is the lowest result. However, the SOM-GCS model performed best with varying data sizes. Similarly, with DailyMail/CNN dataset, SOM-GCS has a fscore value of 95.76%. In contrast, HIBERT models have a fscore of 91.45%, the second highest number, and DSNsum has a fscore of 74.65%, the lowest value for ROUGE 1. Similarly, with ROUGE 2, the proposed method SOM-GCS has a fscore of 97.12% and 97.89% for DUC 2004 and DailyMail/CNN dataset respectively, while it is between 85.13% and 96.77% for the other existing methods for both DUC 2004 and DailyMail/CNN dataset. For ROUGE-L, the proposed SOM-GCS resulted in an improved performance with fscore. However, the SOM-GCS model has shown maximum performance with DUC 2004 data set. The SOM-GCS has 93.44% of fscore, while it is 91.89%, 92.23%, 87.87%, 83.89%, and 88.87% for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models, respectively for ROUGE L. And with DailyMail/CNN dataset SOM-GCS model has shown maximum performance of fscore value with 94.87%. In comparison, it is 89.13%, 92.15%, 88.97%, 84.13%, and 88.97% for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models, respectively.

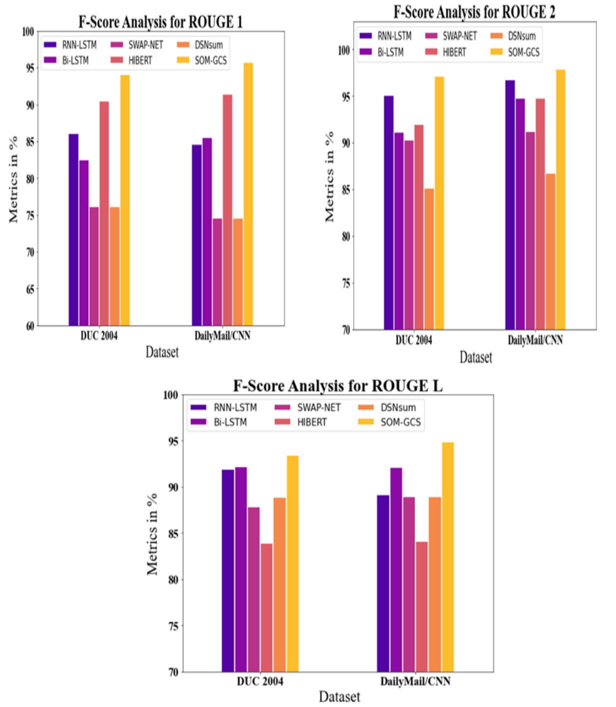

Fig 7: F-Score Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

4.3.5. Matthew's correlation coefficient (MCC) Analysis

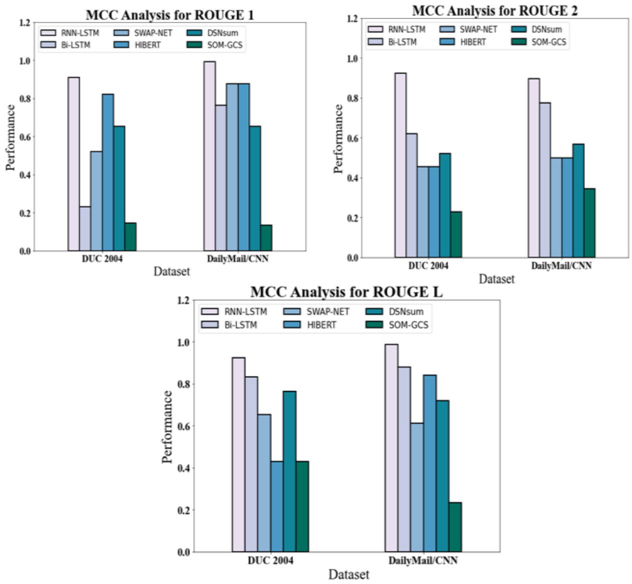

Fig 8: MCC Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Fig.8 and Tab.5 show an mcc comparison of the SOM-GCS methodology with other available approaches. The graph illustrates that SOM-GCS for ROUGEs resulted in an improved performance with mcc. For example, with dataset DUC 2004, SOM-GCS has an MCC value of 0.145, whereas HIBERT models have an MCC of 0.823, and Bi-LSTM has an mcc of 0.231. However, the SOM-GCS model performed best with varying data sizes. Similarly, with DailyMail/CNN dataset, SOM-GCS has an MCC value of 0.134, whereas SWAP-NET models have an MCC of 0.876, and Bi-LSTM has an mcc of 0.765 for ROUGE 1. Similarly, with ROUGE 2, the proposed method SOM-GCS has an MCC of 0.228 and 0.354 for DUC 2004 and DailyMail/CNN dataset respectively. It is between 0.456 and 0.897 for the other existing methods for both DUC 2004 and DailyMail/CNN datasets. For ROUGE-L, the SOM-GCS model has shown maximum performance with DUC 2004 data set. The MCC value of SOM-GCS is 0.765, while it is 0.923, 0.834, 0.654, 0.431, and 0.765 for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models, respectively. And with DailyMail/CNN dataset SOM-GCS model has shown maximum performance for MCC value with 0.234, while it is 0.987, 0.879, 0.611, 0.841, and 0.721 for RNN-LSTM, Bi-LSTM, SWAP-NET, HIBERT, and DSNsum models respectively.

5. Conclusion

This article's main topic is the ATS (Automatic document summarizing) methods that have recently been introduced. Recent advances in text summarization have been made possible by using deep learning-based models for multi-document summarizing. These models also enhance model performance. We propose the Self-Organizing Maps-based Graph Convolutional Summarizer (SOM-GCS) as an alternative to the standard approaches, as it is an extractive multi-document summary method that uses SOM to guarantee a minimum performance restriction. It corrects the faults of SMO-GCS and makes improvements that result in a summarizer that supports phrase embedding and feature learning while considering the graph structure. It is necessary to use a rigorous methodology to show how enhancements are feasible while still ensuring a low-performance restriction. The proposed summarization approach is tested on the DUC 2004 and DailyMail/CNN datasets. The new consequences establish that, in terms of ROUGE scores, SOM-GCS performs on par with cutting-edge summarization techniques. We tested this summarizer using popular datasets, comparing its results to some of the most advanced summarizers. These tests demonstrated that the suggested summarizer performs better than others. There are numerous further applications, including feature selection. In our upcoming study, we want to investigate various similarity techniques, like neural network-based similarity models for summary, which might function well and impact the effectiveness of the summarizer.

References

[1] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. 2018. Big Data Technologies: A Survey. Journal of King Saud University-Computer and Information Sciences 30, 4, 431–448.

[2] Gangathimmappa, M, Subramani, N, Sambath, V, Ramanujam, RAM, Sammeta, M. Deep learning enabled cross-lingual search with metaheuristic web-

based query optimization model for multi-document summarization. Concurrency Computation Practice Experience. 2022;e7476. doi:10.1002/cpe.7476

[3] Abbas Mardani, Sudhanshu Maurya & N. Arulkumar (2023) Eagle Strategy Arithmetic Optimisation Algorithm with Optimal Deep Convolutional Forest Based FinTech Application for Hyper-automation, Enterprise Information Systems, DOI: 10.1080/17517575.2023.2188123

[4] Ghadimi, A., & Beigy, H. (2020). Deep submodular network: An application to multi-document summarization. Expert Systems with Applications, 152, 113392.

[5] Shanmugavadivel, K., Sathishkumar, V.E., Neelakandan.S. et al. Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Sci Rep 12, 21557 (2022). https://doi.org/10.1038/s41598-022-26092-3

[6] Widyassari PA, Afandy NE, Fanani AZ, Syukur A, Basuki RS (2019) Literature review of automatic text summarization: research trend, dataset and method. In: IEEE International conference on information and communications technology (ICOIACT), pp 491–496

[7] Ranabhat R, Upreti A, Sangpang B, Manandhar S (2019) Salient sentence extraction of Nepali online health news texts. Int J Adv Soc Sci, pp 21–26.

[8] Moratanch N, Chitrakala S (2017) A survey on extractive text summarization. In: 2017 International conference on computer, communication and signal processing (ICCCSP), pp 1–6

[9] Kumar, Y., Kaur, K., & Kaur, S. (2021). Study of automatic text summarization approaches in different languages. Artificial Intelligence Review, 54(8), 5897-5929.

[10] Reinald Kim Amplayo and Mirella Lapata. 2019. Informative and Controllable Opinion Summarization. arXiv preprint arXiv:1909.02322

[11] Abhishek Kumar Singh, Manish Gupta, and Vasudeva Varma. 2018. Unity in Diversity: Learning Distributed Heterogeneous Sentence Representation for Extractive Summarization. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018). New Orleans, United States, 5473–5480.

[12] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring Sentence Singletons and Pairs for Abstractive Summarization. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019). Florence, Italy, 2175–2189.

[13] Bahloul B, Aliane H, Benmohammed M (2019) ArA*summarizer: An Arabic text summarization system based on subtopic segmentation and using an A* algorithm for reduction. Wiley Expert systems, New York, pp 1–16.

[14] S. V. Mokhale and G. M. Dhopawkar, "A Study on Different Multi-Document Summarization Techniques," 2019 Third International Conference on Inventive Systems and Control (ICISC), 2019, pp. 710-713, doi: 10.1109/ICISC44355.2019.9036387.

[15] M. Y. Saeed, M. Awais, R. Talib and M. Younas, "Unstructured Text Documents Summarization With Multi-StageClustering," in IEEE Access, vol. 8, pp. 212838-212854, 2020, doi: 10.1109/ACCESS.2020.3040506.

[16] S. U. T. L. ENCODER, "Deep learning based abstractive Arabic text summarization using two layers encoder and one layer decoder," Journal of ?eoretical and Applied Information Technology, vol. 98, no. 16, 2020

[17] Ukan, T., & Karc, A. (2020). Extractive multi-document text summarization based on graph independent sets. Egyptian Informatics Journal. doi:10.1016/j.eij.2019. 12.002.

[18] Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-Granularity Interaction Network for Extractive and Abstractive Multi-Document Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). Online, 6244–6254.

[19] Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 2017. Cascaded Attention based Unsupervised Information Distillation for Compressive Summarization. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). Copenhagen, Denmark, 2081–2090.

[20] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving Multi-document Summarization via Text Classification. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017). San Francisco, United States, 3053–3059.

[21] C. Zhuang, Y. Zheng, W. Huang, and W. Jia, "Joint Fine-Grained Components Continuously Enhance Chinese Word Embeddings," IEEE Access, vol. 7, pp. 174699 174708, 2019, doi: 10.1109/ACCESS.2019.2956822

[22] D. Zhao, J. Wang, Y. Chu, Y, Zhang, Z. Yang, and H. Lin, "Improving biomedical word representation with locally linear embedding,"Neurocomputing, vol. 447, pp. 172-182, Aug. 2021, doi:10.1016/j.neucom.2021.02.071.

[23] E. F. Ayetiran, P. Sojka, and V. Novonty, "EDS-MEMBED:Mulyi-sense embeddings based on enhanced distributional semantic structures via a graph walk over word senses,"Knowledge-Based Systems, vol. 219, pp. 106902, May 2021, doi:10.1016/j.knosys.2021.106902.

[24] C. Mallick, A. K. das, W. Ding, and J. Nayak, "Ensemble summarization of bio-medical articles

integrating clustering and multi-objective evolutionary algorithms,” Applied Soft Computing, vol. 106, p. 107347, Jul. 2021, doi: 10.1016/j.asoc.2021.107347.

[25] Alguliyev, RM, Aliguliyev, RM, Isazade, NR, Abdi, A & Idris, N 2019, 'COSUM: Text summarization based on clustering and optimization', Expert Systems, vol. 36, no. 1.

[26] Rautray, R., & Balabantaray, R. C. (2017). Cat swarm optimization based evolutionary framework for multi document summarization. Physica a: statistical mechanics and its applications, 477, 174-186.

[27] Yin, H. (2008). The self-organizing maps: background, theories, extensions and applications. Computational intelligence: A compendium, 715-762.

[28] D. Paulraj, P. Ezhumalai & M. Prakash (2022) A Deep Learning Modified Neural Network(DLMNN) based proficient sentiment analysis technique on Twitter data, Journal of Experimental & Theoretical Artificial Intelligence, DOI: 10.1080/0952813X.2022.2093405

[29] Arun, A., Bhukya, R. R., Hardas, B. M., Ch., T. et al. (2022). An Automated Word Embedding with Parameter Tuned Model for Web Crawling. Intelligent Automation & Soft Computing, 32(3), 1617–1632.

Table 1. Precision Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Methods	**DUC 2004**			**DailyMail/CNN**		
	ROUGE 1	ROUGE 2	ROUGE L	ROUGE 1	ROUGE 2	ROUGE L
RNN-LSTM	87.90	76.97	79.34	86.57	77.65	80.54
Bi-LSTM	78.97	81.87	82.54	75.05	83.76	83.43
SWAP-NET	90.75	91.67	90.45	91.87	89.06	93.76
HIBERT	79.67	74.98	78.34	76.45	75.39	79.03
DSNsum	83.54	69.80	84.32	85.17	73.67	85.96
SOM-GCS	93.06	96.05	96.25	94.56	96.05	97.32

Table 2. Recall Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Methods	**DUC 2004**			**DailyMail/CNN**		
	ROUGE 1	ROUGE 2	ROUGE L	ROUGE 1	ROUGE 2	ROUGE L
RNN-LSTM	76.90	75.90	76.93	80.43	81.67	82.34
Bi-LSTM	89.34	89.21	89.12	89.45	89.78	90.65
SWAP-NET	82.34	82.68	82.34	84.56	84.98	85.01
HIBERT	88.34	85.90	86.43	84.78	88.76	89.64
DSNsum	74.21	76.54	77.93	80.94	79.13	80.14
SOM-GCS	91.28	93.87	95.89	92.89	94.76	96.19

Table 3. Classification Accuracy Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Methods	**DUC 2004**	**DailyMail/CNN**

	ROUGE 1	ROUGE 2	ROUGE L	ROUGE 1	ROUGE 2	ROUGE L
RNN-LSTM	76.98	78.38	79.34	77.84	80.14	75.83
Bi-LSTM	82.19	82.67	83.15	82.75	83.48	84.95
SWAP-NET	71.54	72.78	74.38	71.98	74.67	75.13
HIBERT	89.34	91.87	93.61	90.65	92.59	94.45
DSNsum	85.76	86.63	86.54	86.43	85.38	87.12
SOM-GCS	96.76	97.56	98.24	96.85	96.79	98.95

Table 4. F-Score Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Methods	**DUC 2004**			**DailyMail/CNN**		
	ROUGE 1	ROUGE 2	ROUGE L	ROUGE 1	ROUGE 2	ROUGE L
RNN-LSTM	86.13	95.13	91.89	84.65	96.77	89.13
Bi-LSTM	82.55	91.13	92.23	85.51	94.77	92.15
SWAP-NET	76.13	90.34	87.87	74.67	91.21	88.97
HIBERT	90.56	91.98	83.89	91.45	94.77	84.13
DSNsum	76.15	85.13	88.87	74.65	86.77	88.97
SOM-GCS	94.12	97.12	93.44	95.76	97.89	94.87

Table 5. MCC Analysis of SOM-GCS for ROUGE-1, ROUGE-2, and ROUGE-L

Methods	**DUC 2004**			**DailyMail/CNN**		
	ROUGE 1	ROUGE 2	ROUGE L	ROUGE 1	ROUGE 2	ROUGE L
RNN-LSTM	0.911	0.923	0.923	0.994	0.956	0.987
Bi-LSTM	0.231	0.621	0.834	0.765	0.897	0.879
SWAP-NET	0.521	0.456	0.654	0.876	0.498	0.611
HIBERT	0.823	0.456	0.431	0.876	0.498	0.841
DSNsum	0.654	0.521	0.765	0.653	0.567	0.721
SOM-GCS	0.145	0.228	0.765	0.134	0.345	0.234